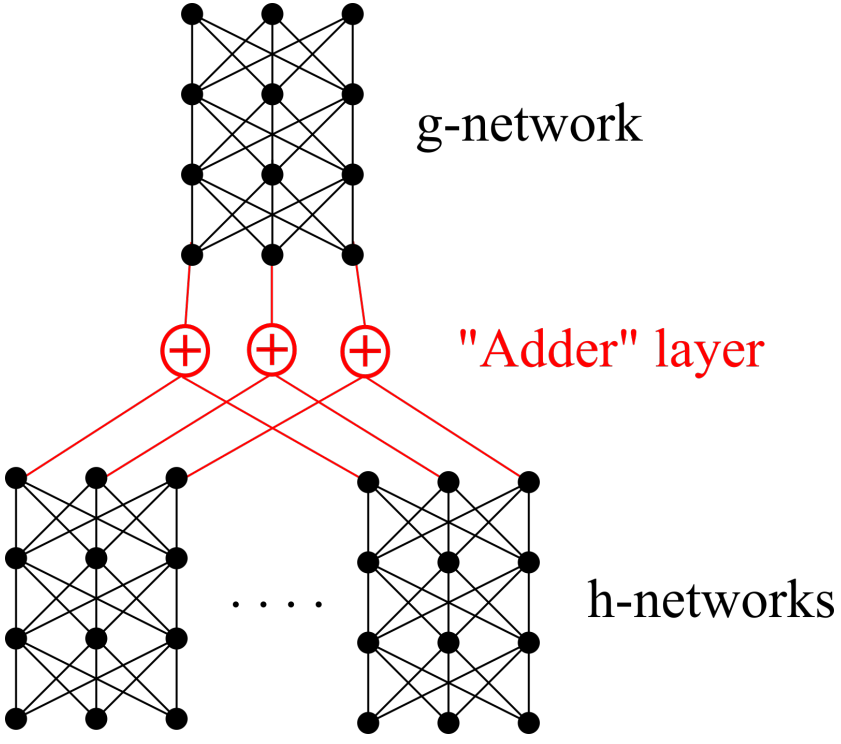
Symmetric neural networks

- Permutation invariance can be handled by symmetric neural networks
- I wasted 2 years trying to solve this problem, and then find out that it has been solved 3 years ago: [PointNet 2017] and [DeepSets 2017] and their mastery of mathematics is significantly above me!
- Any symmetric function can be represented by the following form (a special case of the Kolmogorov-Arnold representation of functions):

$$f(x, y, ...) = g(h(x) + h(y) + ...) \tag{7}$$



(8)

- Sym NN gives a powerful boost in efficiency $\propto n!$ where $n = \#\text{inputs}$
- The code for Sym NN is just a few lines of Tensorflow:

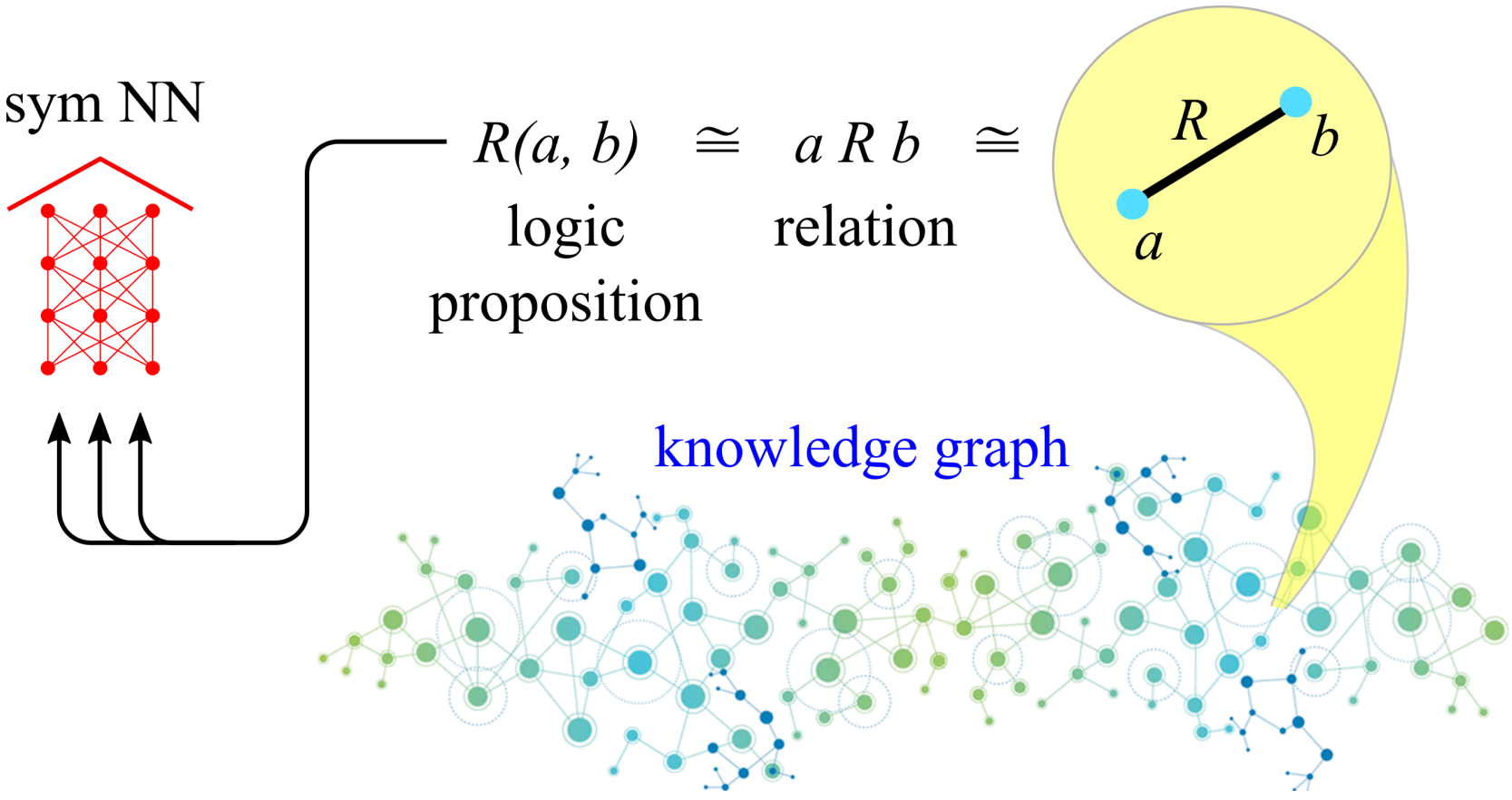
```
h = Dense(3, activation='tanh')
ys = []
for i in range(9):
    ys.append( h(xs[i]) )
y = Keras.stack(ys, axis=1)
Adder = Lambda(lambda x: Keras.sum(x, axis=1))
y = Adder(y)
g = Dense(3)
output = g(y)
```

(9)

- Very easy to adopt this to existing models such as BERT and reinforcement learning
- I have successfully tested it on the game of TicTacToe

Application: knowledge graphs

- One cannot feed a knowledge graph directly into an NN, as its input must be embedded in vector space. A solution is to break the graph into edges, where each edge is equivalent to a relation or proposition. One could say that graphs are isomorphic to logic

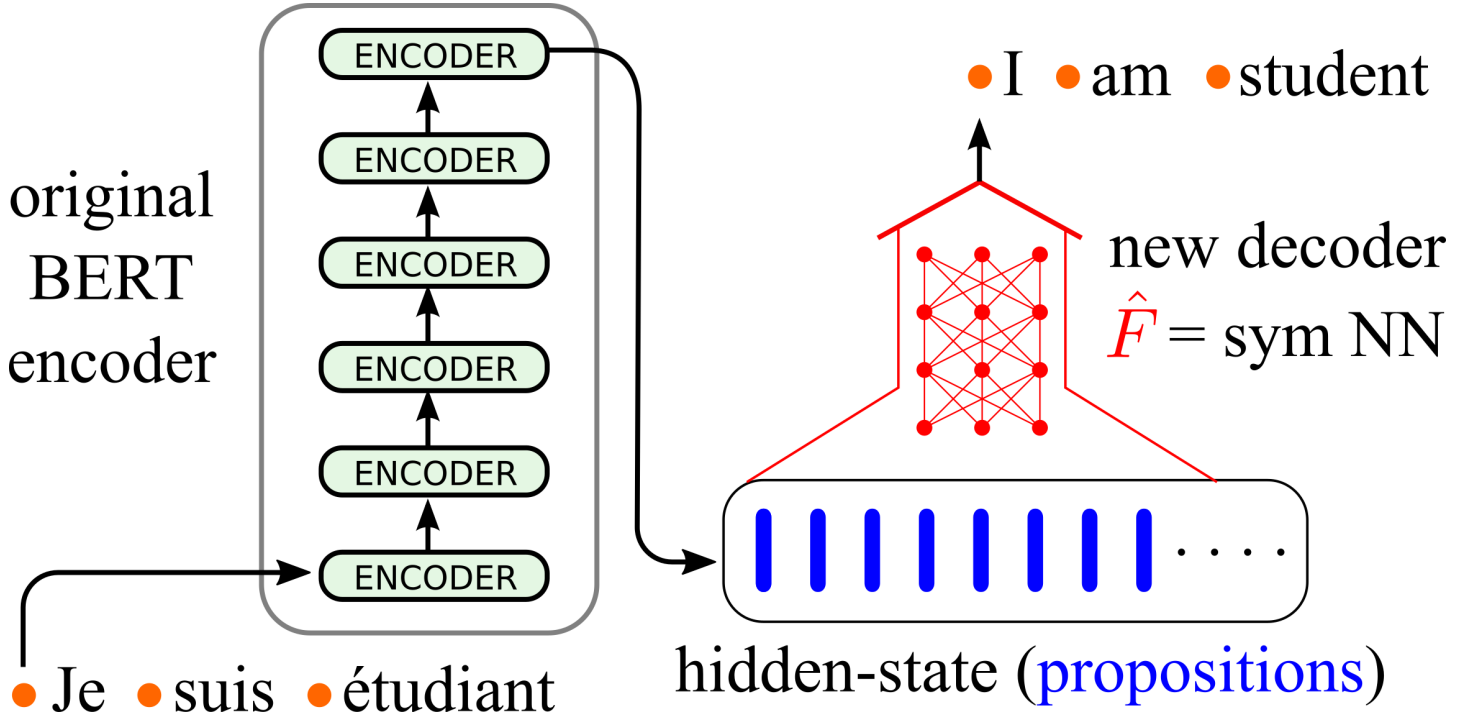


(10)

- Since edges are invariant under permutations, it appears that symmetric NNs are required to process them

Application: logicalization of BERT

- Similarly, we can convert BERT's hidden state into a set of propositions, by replacing the original decoder with a sym NN:

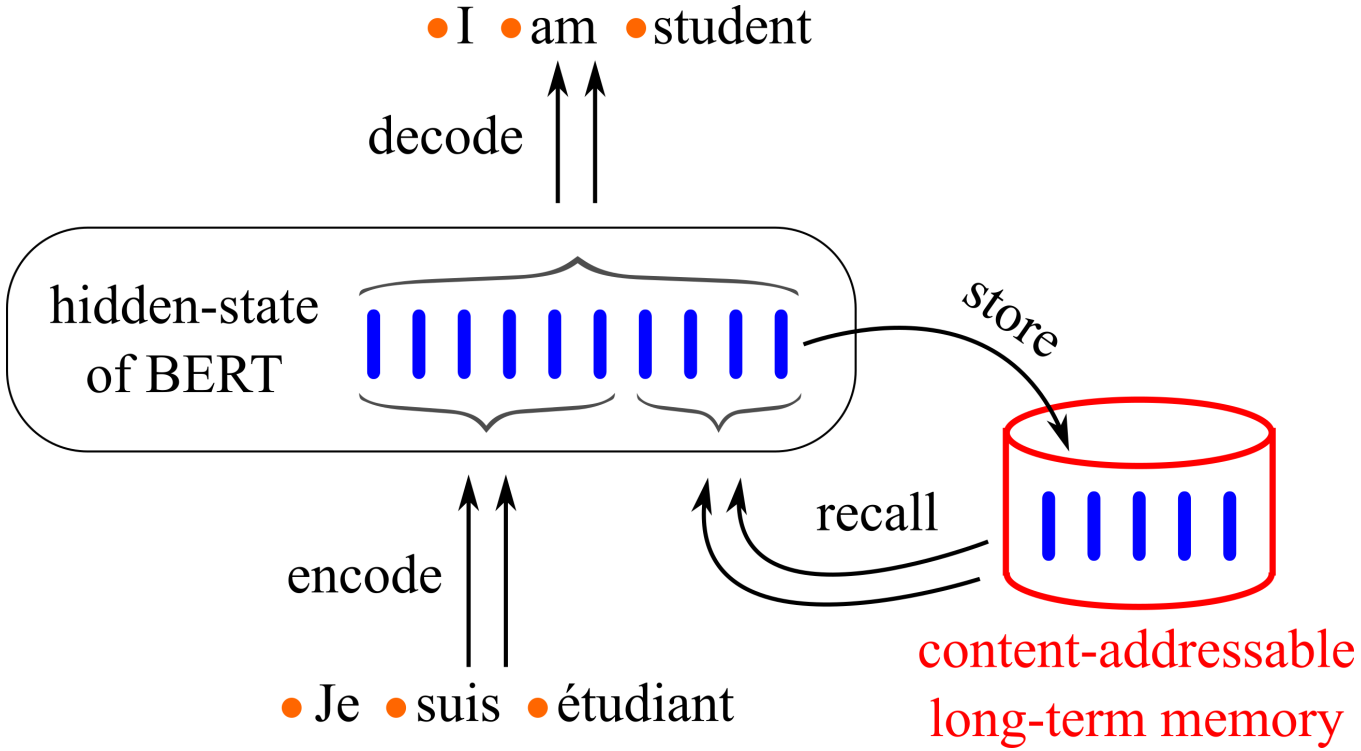


(11)

- The original encoder can be retained. As the decoder imposes symmetry on the hidden state, error propagation is expected to cause its representation to change
- Of course, this remains to be proven by experiment ☺

Application: content-addressable long-term memory

- The original BERT hidden state lacked a logical structure and it was not clear what exactly it contains. After logicalization, propositions inside BERT can be stored into long-term memory:



(12)

- These kind of systems are very close to strong AI, and it depends crucially on logicalization
- The content-addressable memory idea came from Alex Graves *et al*'s Neural Turing Machine [2014]